SOFT MAIL

NO.EXE
โปรแกรมยกเว้นการเรียกใช้ไฟล์

เพิ่มประสิทธิภาพของการใช้คำสั่ง COPY, DEL, DIR, RENAME ให้สะดวกขึ้น

ผมคิดว่าคุณคงเคยต้องการลบไฟล์เกือบทั้งหมดบนดิสก์ โดยยกเว้นเพียงบางไฟล์เท่านั้น เช่น ต้องการลบไฟล์ทุกไฟล์บนดิสก์ ยกเว้น *.ASM เป็นต้น ซึ่งคุณไม่สามารถใช้คำสั่ง DEL *.* แน่ กรณีนี้ทำได้วิธีเดียวคือลบไฟล์ออกทีละนามสกุล

NO.EXE ผู้ช่วยคนใหม่

ปัญหาที่ผมยกขึ้นมาในตอนต้นนั้น NO.EXE จะช่วยคุณได้ เพียงแต่คุณใช้คำสั่งนี้

NO *.ASM DEL *.*

หลังจากคุณใช้คำสั่งดังกล่าวไฟล์ทุกไฟล์บนดิสก์จะถูกลบออกทันทียกเว้นไฟล์ที่มีนามสกุล .ASM เท่านั้น นอกจากนี้ยังใช้ NO.EXE กับคำสั่ง DIR, COPY และ RENAME เช่น ถ้าต้องการก๊อบปี้ไฟล์ทุกไฟล์ยกเว้น *.COM ไปที่ไดรฟ์ A: ก็ทำได้โดยใช้คำสั่ง

NO *.COM COPY *.* A:

การใช้ NO.EXE มีรูปแบบคำสั่งง่าย ๆ ดังนี้

NO filespec Command (parameter) นั้นขึ้นกับคำสั่งที่ใช้ซึ่งจะใส่หรือไม่ก็ได้ เช่น DIR ไม่จำเป็นต้องมี Parameter ก็ได้ เป็นต้น ส่วน filespec ในที่นี้คุณสามารถระบุไดรฟ์ หรือ Path รวมทั้งจะใช้ * และ ? ด้วยก็ได้ ข้อสำคัญก็คือ พารามิเตอร์แต่ละตัวของ NO.EXE ต้องแยกจากกันด้วยช่องว่างอย่างน้อยหนึ่งตัว

กรณีที่คุณต้องการยกเว้นมากกว่าหนึ่งกลุ่ม คุณก็ทำได้โดยเรียกใช้ NO .EXE ซ้อนกันได้ เช่น ถ้าต้องการลบไฟล์ทุกไฟล์ ยกเว้น *.ASM และ *.COM ก็ทำได้โดยคำสั่ง

NO *.ASM NO *.COM DEL *.*

ในที่นี้ Command คือ NO และ Parameter คือ *.COM DEL *.* เป็นต้น

NO.EXE ทำงานอย่างไร

คำสั่งของดอสส่วนใหญ่นั้นจะไม่ทำงานกับไฟล์ที่มีแอตทริบิวต์แบบ Hidden ซึ่งจากตารางที่ 1 จะเห็นว่าถ้าบิตที่ 2 ของแอตทริบิวต์ถูกเซตก็จะทำให้ไฟล์นั้นกลายเป็นไฟล์แบบ Hidden

ค่าแอตทริบิวต์	ความหมาย
01h	แสดงว่าเป็นไฟล์ที่อ่านได้อย่างเดียว (read-only)
02h	แสดงว่าเป็นไฟล์ที่ถูกซ่อนไว้ (hidden)
04h	แสดงว่าเป็นซิสเต็มไฟล์
08h	แสดงว่าเป็นชื่อโวลุมเลเบล (volume label)
10h	แสดงว่าเป็นชื่อสับไดเรกตอรี่
20h	แสดงว่าเป็น archive bit ซึ่งจะใช้โดย BACKUP และ RESTORE

ตารางที่ 1 แสดงความหมายของแอตทริบิวต์ของไฟล์

ทันที NO.EXE จะทำการเปลี่ยนแอตทริบิวต์ของไฟล์กลุ่มที่จะยกเว้นทั้งหมดให้เป็นแบบ Hidden โดยโพรซิเดอร์ process_file จะค้นหาไฟล์ทั้งหมดด้วยโพรซิเดอร์ findfirst และ findnext จากนั้นก็จะแก้ค่าแอตทริบิวต์ของไฟล์โดยโพรซิเดอร์ change_attr แล้วจึงเรียกคอมมานด์โพรเซสเซอร์ชุดที่ 2 มาทำงานตามคำสั่งที่คุณต้องการแต่จุดสำคัญอยู่ที่ต้องค้นหา Path ที่จะเข้าถึง COMMAND.COM (ถูกกำหนดไว้ในตัวแปร COMSPEC) ก่อน สุดท้ายก็เปลี่ยนแอตทริบิวต์ไฟล์กลุ่มที่ยกเว้นกลับเป็นค่าเดิม

การค้นหา COMSPEC

ปกติแล้วดอสจะเก็บค่าของ COMSPEC ไว้ภายในเอ็นไวรอนเมนต์ของโปรแกรม ซึ่งค่าเซกเมนต์แอดเดรสของเอ็นไวรอนเมนต์จะเก็บอยู่ที่ออฟเซตแอดเดรส 2Ch ภายใน Program segment prefix(PSP) โดยทั่วไปไฟล์ที่มีนามสกุล .EXE นั้นเมื่อถูกโหลดลงบนหน่วยความจำครั้งแรก ค่าของรีจิสเตอร์ DS และ ES จะชี้ไปที่ค่าแอดเดรสของ PSP ทันที แต่เนื่องจากโปรแกรม NO.EXE ถูกเขียนด้วยเทอร์โบปาสคาล ซึ่งเทอร์โบจะแก้ค่าของ DS และ ES ใหม่ให้ชี้ไปยังดาต้าเซกเมนต์และเอกซ์ตร้าเซกเมนต์ทันทีก่อนที่จะทำงานตามโปรแกรมที่เราเขียน ดังนั้นจึงไม่สามารถใช้ฟังก์ชัน dseg เพื่ออ่านค่า DS มาใช้เป็นแอดเดรสของ PSP ได้

เนื่องจากดอสจะโหลดโปรแกรมบนหน่วยความจำต่อจาก PSP ทันที เราจึงใช่ค่าโค้ดเซกเมนต์แอดเดรส มาหาค่าแอดเดรสของ PSP ได้ โดยที่เราทราบว่าขนาดของ PSP เท่ากับ 256 ไบต์ (10h พารากราฟ) เพราะฉะนั้นถ้าอ่านค่าจากฟังก์ชัน cseg มาลบด้วย 10h ก็จะได้ค่าแอดเดรสของ PSP จากนั้นก็นำเอาเอ็นไวรอนเมนต์แอดเดรสมาหาค่าของ COMSPEC

```
A>DEBUG
-A
XXXX:0100 MOV SI,80
XXXX:0103 MOV BL,[SI]
XXXX:0105 XOR BH,BH
XXXX:0107 MOV BYTE PTR [BX+SI+1],0
XXXX:010B MOV DX,82
XXXX:010E MOV AX,4300
XXXX:0111 INT 21
XXXX:0113 AND CL,FD
XXXX:0116 MOV AX,4301
XXXX:0119 INT 21
XXXX:011B INT 20
XXXX:011D
-N UNHIDE.COM
-R CX
CX 0000
:1D
-W
Writing 001D bytes
-Q
```

รูปที่ 1
โปรแกรม UNHIDE.COM ที่สร้างด้วย DEBUG.COM

ก่อนจบ

เนื่องจาก NO.EXE มีการเปลี่ยนแอตทริบิวต์ของไฟล์เป็นแบบ Hidden ดังนั้นหากมีการหยุดทำงานของโปรแกรมขณะที่โปรแกรมยังทำงานไม่เสร็จ เช่น ไฟดับ ไฟล์ที่ถูกเปลี่ยนแอตทริบิวต์ไปจะไม่สามารถเรียกดูด้วยคำสั่ง DIR ซึ่งคุณอาจใช้ PCTOOLS เปลี่ยนแอตทริบิวต์กลับ หรือสร้างโปรแกรมตามรูปที่ 1 โดยถ้าคุณใช้ CHKDSK/V คุณก็จะได้รายชื่อไฟล์ที่ซ่อนอยู่ จากนั้นจึงใช้ UNHIDE.COM ช่วย โดยใช้คำสั่ง UNHIDE ตามด้วยชื่อไฟล์ หวังว่า NO.EXE คงช่วยให้การทำงานของคุณสะดวกขึ้น ส่วนซอร์สโปรแกรมแสดงดังรูปที่ 2 ครับ

รูปที่ 2 ซอร์สโปรแกรม NO.PAS ที่เขียนด้วยเทอร์โบปาสคาล

```
{*********************************************************}
{*  NO.PAS do exclude some file from the access of DOS.  *}
{*  Syntax :                                             *}
{*           NO [filespec] Command [parameter]           *}
{*********************************************************}
program no_access;
{$M $4000, 0, 0}
  uses dos, crt;
  type str_type = string[80];
  var  file_spec,
       full_name,
       path_name,
       cmd_path,
       command    : str_type;
  function getdisk : char;
    var   regs : registers;
  begin
    regs.ah := $19;
    msdos(regs);
    getdisk := char(regs.al + byte('A'))
  end;
  procedure get_param;
    var    i : byte;
  begin
    file_spec := paramstr(1);
    command := concat('/C ',paramstr(2));
    for i := 3 to paramcount do
    begin
      command := concat(command,' ');
      command := concat(command,paramstr(i))
    end
  end;
  procedure get_cmd_path(var cmd_path : string);
    var    env_seg, psp_seg : word;
           ok               : boolean;
           i, j             : byte;
           comspec, s       : string[8];
  begin
    psp_seg := cseg - $10;
    env_seg := memw[psp_seg:$2C];
    i := 0;
    ok := false;
    comspec := 'COMSPEC=';
    while not ok do
    begin
      if char(mem[env_seg:i]) = 'C' then
      begin
        s := '';
        for j := i to i + 8 do
          s := concat(s,char(mem[env_seg:j]));
        if s = comspec then
          ok := true;
        i := j
      end;
      inc(i)
```

```
    end;
    dec(i);
    cmd_path := '';
    while mem[env_seg:i] <> 0 do
    begin
       cmd_path := concat(cmd_path,char(mem[env_seg:i]));
       inc(i)
    end
 end;
 procedure set_full_name;
 begin
    if pos(file_spec,'\') = 0 then
    begin
       getdir(0,full_name);
       if full_name[length(full_name)] <> '\' then
          full_name := concat(full_name,'\');
       if file_spec[2] = ':' then
          full_name := file_spec
       else
          full_name := concat(full_name,file_spec)
    end
    else
       begin
          if file_spec[2] <> ':' then
          begin
             full_name[1] := getdisk;
             full_name[2] := ':';
             if file_spec[1] <> '\' then
                full_name[3] := '\'
          end;
          full_name := concat(full_name,file_spec)
       end
 end;
 procedure set_path_name;
    var    i : byte;
 begin
    i := length(full_name);
    while (full_name[i] <> ':') and (full_name[i] <> '\') do dec(i);
    path_name := copy(full_name,1,i)
 end;
 procedure error(errno : byte);
    var    msg : string[40];
 begin
    case errno of
       0 : msg := 'Syntax: NO FILESPEC COMMAND [PARAMETERS]';
       2 : msg := 'File or directory not found';
       8 : msg := 'Not enough memory';
      10 : msg := 'Invalid environment';
      11 : msg := 'Invalid format'
      else msg := ''
    end;
    writeln(msg)
 end;
 procedure process_file(mode : byte);
    var    dta : searchrec;
    procedure change_attr;
```

```
      var   f           : file;
            p           : str_type;
            a, a_b, o_b : byte;
    begin
      case mode of
        0 : begin
              a_b := $FF;
              o_b := $02
            end;
        1 : begin
              a_b := $FD;
              o_b := $00
            end
      end;
      a := dta.attr and a_b;
      a := a or o_b;
      p := concat(path_name,dta.name);
      assign(f,p);
      setfattr(f,a)
    end;
  begin
    findfirst(full_name,anyfile,dta);
    while doserror = 0 do
    begin
      change_attr;
      findnext(dta)
    end;
    if doserror <> 0 then error(doserror)
  end;
  procedure process_cmd;
  begin
    get_cmd_path(cmd_path);
    exec(cmd_path,command);
    if doserror <> 0 then error(doserror)
  end;
begin
  if paramcount > 0 then
  begin
    checkbreak := false;
    get_param;
    set_full_name;
    set_path_name;
    process_file(0);
    process_cmd;
    process_file(1)
  end
  else
    error(0)
end.
```